

```
def SLvaDS(lst):  
    print('Số lượng các con thú:', len(lst))  
    print('Danh sách các con thú:', lst)
```


```
def TimKiem(lst):  
    ten = input('Nhập tên con thú cần tìm')  
    if ten in lst:  
        print('%s có trong lst' % (ten))  
    else:  
        print('%s không có trong lst' % (ten))
```

```
lst = ['ant', 'bear', 'cat', 'dog', 'elephant', 'fish', 'goat', 'hippo']  
SLvaDS(lst)  
TimKiem(lst)
```


10/. Viết chương trình gồm các hàm (function) để thực hiện các chức năng như sau:

a. Cho người dùng nhập nhiều lần các số nguyên dương. Việc nhập sẽ kết thúc khi người dùng nhập số âm. Đưa tất cả các số đã nhập (không kể số âm nhập cuối cùng) vào list L. Hàm trả về list L.


b. Hàm trả về tổng các số có trong List.

 Gợi ý: sử dụng hàm sum(listName) để tính tổng các số có trong list.

c. Hàm nhận tham số là list L và số nguyên X. Tìm xem x có trong list chứa các số vừa nhập hay không? Nếu có, cho biết x xuất hiện bao nhiêu lần?

 Gợi ý: sử dụng phương thức count của đối tượng list (listName.count(x)).

d. Hàm nhận tham số là list L và số nguyên x. Cho biết X có lớn hơn tất cả các số có trong list hay không? Nếu không, hãy in ra các số có trong list và lớn hơn x.

 Gợi ý: sử dụng hàm `max(listName)` để tìm số lớn nhất trong `listName`.

e. Hàm trả về tổng khoảng cách giữa tất cả các cặp số có trong L.

Ví dụ: list `L = [1, 2, 3]` in ra 'Tổng khoảng cách giữa các số là: 4'

Giải thích: vì  $|1-2| + |2-3| + |1-3| = 1 + 1 + 2 = 4$

```
def NhapSo():  
    lst = []  
    while True:  
        try:  
            n = int(input("Vui long nhap so nguyen duong n: "))  
        except Exception:  
            print("gia tri ko phai kieu so")  
        else:  
            if type(n) is not int:  
                print("vui long nhap so nguyen duong")  
            elif n < 0:  
                break  
            else:  
                print("them so", n, "vao list")  
                lst.append(n)  
    return lst
```

```
def KiemTraX(lst, X):  
    if X >= max(lst):  
        print('Không có số nào lớn hơn %d' % (X))  
    else:  
        for i in lst:  
            if i > X:  
                print(i, end=' ')
```

```
def Dem_x(lst, x):  
    if lst.count(x) > 0:  
        print('%d xuất hiện %d lần trong lst' % (x, lst.count(x)))  
    else:  
        print('%d không xuất hiện trong lst' % (x))
```

```
- def TongKC(lst):  
    Tong = 0  
    for i in range(0, len(lst)-1):  
        for j in range(i+1, len(lst)):  
            Tong = Tong + abs(lst[i]-lst[j])  
    return Tong
```

# a

# lst = NhapSo()

# print("List duoc tao:",lst)

# b

# tong = 0

# for i in lst:

# tong += i

# print("Tong cac so trong list: ", tong)

# print("Tong cac so trong list: ", sum(lst))



```
38 # c
39 # lst = NhapSo()
40 # print("List duoc tao:",lst)
41 # x = int(input("Vui long nhap so nguyen x: "))
42 # count = 0
43 # for i in lst:
44 #     if i == x:
45 #         count += 1
46 # print("so lan xuat hien cua %d: %d lan"%(x,count))
47
48 # d
49 # lst = NhapSo()
50 # print("List duoc tao:",lst)
51 # x = int(input("Vui long nhap so nguyen x: "))
52 # max = max(lst)
53 # print(max)
54 # if (x > max):
55 #     print ("%d la lon nhat",x)
56 # else:
57 #     print("So lon hon x:",end='')
58 #     for i in lst:
59 #         if i > x:
60 #             print("%d "%i,end='')
```



#### 4.2.9. Xóa phần tử trong list

Có 5 cách xóa phần tử vào *list*:

- Xóa phần tử cuối *list*
- Xóa phần tử đầu tiên trong *list* có giá trị khớp với giá trị chỉ định
- Xóa phần tử bằng hàm *del* và dựa trên *index*
- Xóa toàn bộ *list* bằng hàm *del*
- Xóa rỗng (xóa toàn bộ các phần tử đang có) *list*

##### 4.2.9.1. Xóa phần tử cuối trong list

- Phương thức *list.pop()*: giúp lấy (xóa) phần tử cuối cùng ra khỏi *list*.

- Ví dụ 4.19

<i>Mã lệnh</i>	<i>Kết quả</i>
<code>myList = [1, 2, 3]</code> <code>lastnumber = myList.pop()</code> <code>print (lastnumber)</code>	3
<code>print (myList)</code>	[1, 2]

#### 4.2.9.2. Xóa phần tử đầu tiên trong List có giá trị khớp với giá trị chỉ định

- Phương thức `list.remove(element)`: thực hiện xóa phần tử *element* ra khỏi *list* bằng cách lặp từ đầu đến cuối và so sánh giá trị cần xóa, nếu phần tử nào khớp với giá trị được chỉ định thì xóa.
- Lưu ý
  - (i)- Phương thức chỉ xóa lần so khớp đầu tiên, tức nếu trong list có từ 2 phần tử giống nhau trở lên thì chỉ phần tử đầu tiên bị xóa, các phần tử còn lại sẽ không bị ảnh hưởng.

- Phương thức `list.remove(element)`: thực hiện xóa phần tử *element* ra khỏi *list* bằng cách lặp từ đầu đến cuối và so sánh giá trị cần xóa, nếu phần tử nào khớp với giá trị được chỉ định thì xóa.

- Lưu ý

(i)- Phương thức chỉ xóa lần so khớp đầu tiên, tức nếu trong list có từ 2 phần tử giống nhau trở lên thì chỉ phần tử đầu tiên bị xóa, các phần tử còn lại sẽ không bị ảnh hưởng.

- Ví dụ 4.20

Mã lệnh	Kết quả
<code>aList = [1,2,1,3,1]</code> <code>print ("Before remove:", aList)</code>	Before remove: [1, 2, 1, 3, 1]
<code>aList.remove(1)</code> <code>print ("After remove:", aList)</code>	After remove: [2, 1, 3, 1]

(ii)- Nếu *element* không tồn tại trong list thì chương trình sẽ phát sinh *ValueError exception*.

- Ví dụ 4.21

Mã lệnh	Kết quả
<code>aList.remove(5)</code>	<code>ValueError: list.remove(x): x not in list</code>

''' Cách 3 duyệt list từ cuối về đầu sẽ không gây lỗi và không bị bỏ qua các phần tử nằm kề nhau và cùng thỏa điều kiện '''

```
print('CÁCH 3')
lst=[2,2,3,6,8]
for i in range (len(lst)-1,-1,-1):
    try:
        so=lst[i]
        print('Đang xét index=%d, voi
                gia tri %d' %(i,lst[i]))
        if lst[i] % 2 == 0:
            lst.remove(lst[i])
            print('Xoa %d,list con %d phan
                    tu la:' %(so,len(lst)), lst)
    except IndexError:
        print('IndexError: index=%d, khi
                len(lst)=%d' %(i,len(lst)))
print('List sau xoa:',lst)
```

```
CACH 3
Đang xét index=4, voi gia tri
8
Xoa 8,list con 4 phan tu la:
[2, 2, 3, 6]
Đang xét index=3, voi gia tri
6
Xoa 6,list con 3 phan tu la:
[2, 2, 3]
Đang xét index=2, voi gia tri
3
Đang xét index=1, voi gia tri
2
Xoa 2,list con 2 phan tu la:
[2, 3]
Đang xét index=0, voi gia tri
2
Xoa 2,list con 1 phan tu la:
```

#### 4.2.9.3. Xóa phần tử bằng hàm del và dựa trên index

- Nếu index không hợp lệ sẽ phát sinh lỗi.
- Thứ tự của các phần tử sẽ dịch chuyển tùy vào vị trí của phần tử bị xóa.
- Thực hiện:

- Cách 1: sử dụng **index**

Ví dụ 4.23

Mã lệnh	Kết quả
<pre>chars=['a','b','c','d','e'] del chars[1] print (chars)</pre>	<pre>['a','c','d','e']</pre>

- Cách 2: sử dụng toán tử lấy khoảng **[start:end]**.

Ví dụ 4.24

Mã lệnh	Kết quả
<pre>chars=['a','b','c','d','e'] del chars[1:4] print (chars)</pre>	<pre>['a','e']</pre>



- Thực hiện:

- Cách 1: sử dụng **index**

Ví dụ 4.23

Mã lệnh	Kết quả
<pre>chars=['a','b','c','d','e'] del chars[1] print (chars)</pre>	<pre>['a','c','d','e']</pre>

- Cách 2: sử dụng toán tử lấy khoảng **[start:end]**.


Ví dụ 4.24


Mã lệnh	Kết quả
<pre>chars=['a','b','c','d','e'] del chars[1:4] print (chars)</pre>	<pre>['a','3']</pre>

#### 4.2.9.4. Xóa toàn bộ list ra khỏi bộ nhớ bằng hàm del

- Sau khi xóa list bằng hàm *del*, list sẽ bị xóa khỏi bộ nhớ. Vì vậy, nếu sau đó chương trình có sử dụng *list* sẽ phát sinh lỗi *NameError*.
- Ví dụ 4.25

Mã lệnh	Kết quả
<code>aList = [1,2,1,3,1]</code> <code>print ("Before delete list:", aList)</code>	Before delete list: [1, 2, 1, 3, 1]
<code>del (aList)</code> <code>print ("After delete list:", aList)</code>	<code>NameError: name 'aList' is not defined</code>

 **def** LaSoCP(n):

 **if** **int**(math.sqrt(n))\*\*2==n:

**return True**

**else:**

**return False** 

```
def LaSoMayMan(n):  
    if n<6:  
        return False  
    while n>0:  
        so=n%10  
        if so !=6 and so !=8:  
            return False  
        n=n//10  
    return True
```

# -----

Project ▾

Buoi06 D:\TTTH\_KHTN\Python270

Bai6\_Tr26.py

Bai07\_TR27.py

temp.py


External Libraries

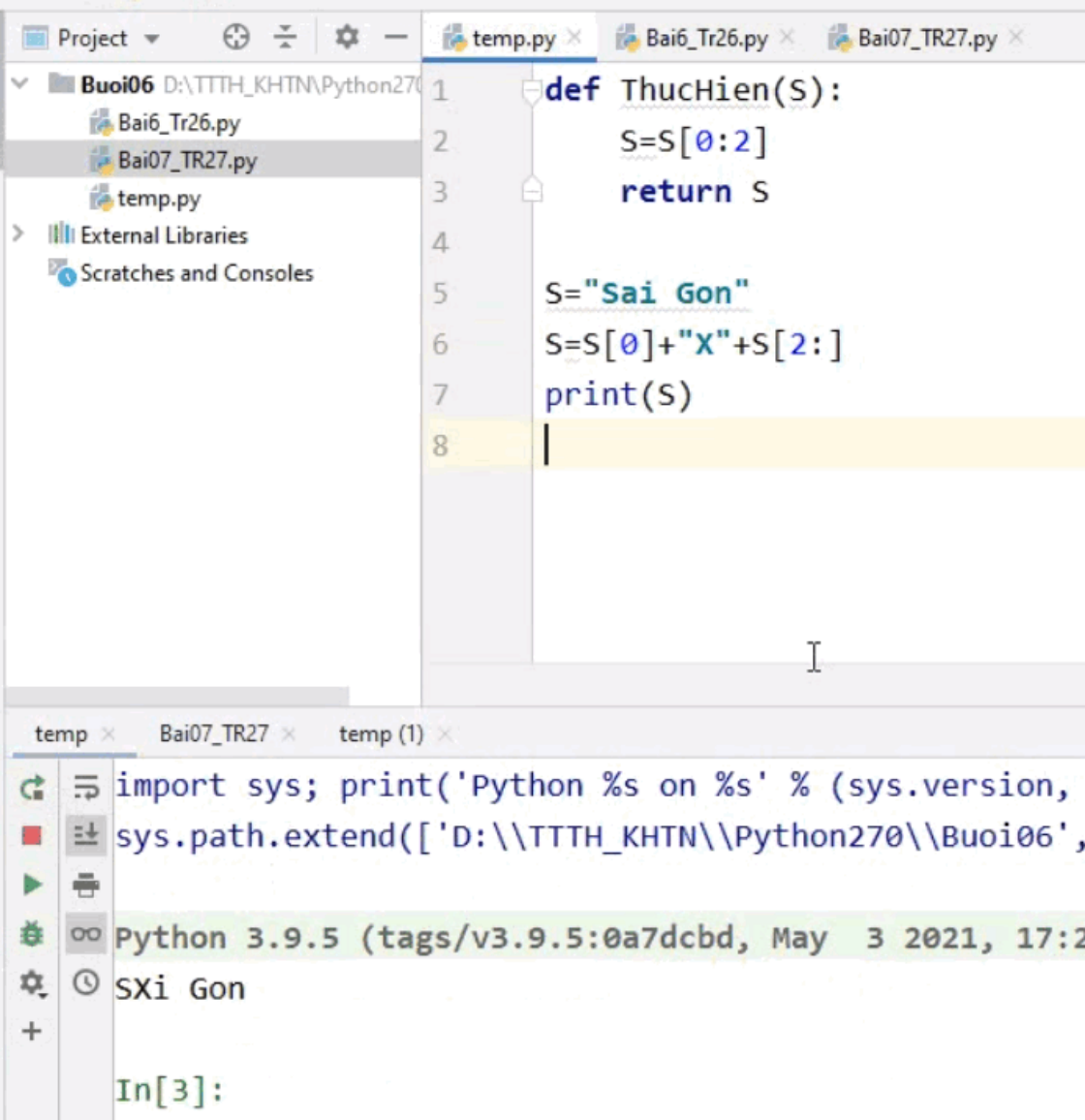
Scratches and Consoles

```
1 def ThucHien(S):  
2     S=S[0:2]  
3     return S  
4  
5     S="Sai Gon"  
6     S=ThucHien(S)  
7     print(S)  
8
```

temp (1) ×

```
import sys; print('Python %s on %s' % (sys.version,  
sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi06'  
Python 3.9.5 (tags/v3.9.5:0a7dcdbd, May 3 2021, 17:  
Sa  
In[3]:
```





64/. Giả sử một chuỗi được gọi là hoàn chỉnh khi:

- Đầu và cuối chuỗi không chứa khoảng trắng (space).
- Giữa các từ chỉ cách nhau bởi 1 khoảng trắng.
- Dấu chấm và dấu phẩy phải đi liền với từ ngay trước mà không được cách bởi khoảng trắng.
- Nếu là bài thơ, các dòng phải được canh thẳng hàng (đều phải xuất phát từ đầu dòng).

Ví dụ: (ký tự `␣` được diễn tả thay cho khoảng trắng)

<i>Chuỗi chưa hoàn chỉnh</i>	<i>Chuỗi hoàn chỉnh</i>
␣␣␣␣Quê hương	Quê␣hương
Quê␣hương␣là␣␣␣chùm␣khế␣␣ngọt.	Quê␣hương␣là␣chùm␣khế␣ngọt.
␣␣␣Cho␣con␣trèo␣hái␣␣␣mỗi␣ngày␣␣.	Cho␣con␣trèo␣hái␣mỗi␣ngày.
Quê␣␣hương␣là␣␣␣đường␣␣đi␣học␣.	Quê␣hương␣là␣đường␣đi␣học.
␣Con␣về␣rợp␣bướm␣vàng␣bay.	Con␣về␣rợp␣bướm␣vàng␣bay.
␣Đỗ␣␣␣␣Trung␣Quân␣␣␣	Đỗ␣Trung␣Quân

Viết chương trình cho người dùng nhập 1 chuỗi (S). Thực hiện xóa tất cả các khoảng trắng thừa.

```
1 def Bai64(S):
2     S=S.strip()
3     while S.find(" ")>0:
4         S=S.replace(" ", " ")
5     while S.find("\n ")>0:
6         S=S.replace("\n ", "\n")
7     while S.find(" .")>0:
8         S=S.replace(" .", ".")
9     return S
10 #----- chương trình chính -----
11 S'''    Quê hương
12 Quê hương là                                |  chòm khế ngọt .
13     Cho con trèo hái    mỗi ngày.
14 Quê    hương là đường đi học .
15     Con về    rợp bướm    vàng bay.
16     Đỗ        Trung Quân    '''
```



Buoi06 > Bai64\_Tr14.py

Project

1: Project

Buoi06

D:\TTTH\_KHTN\Python270

Bai6\_Tr26.py

Bai07\_TR27.py

Bai64\_Tr14.py

temp.py

External Libraries

Scratches and Consoles

temp.py

Bai64\_Tr14.py

Bai6\_Tr26.py

Bai07\_TR27.py

```

7 while S.find( ".") > 0:
8     S=S.replace(" .", ".")
9     return S
10 #----- chương trình chính -----
11 S='''    Quê hương
12    Quê hương là                                chùm khế ngọt .
13        Cho con trèo hái    mỗi ngày.
14    Quê    hương là đường đi học .
15        Con về    rợp bướm    vàng bay.
16    Ở    Trung Quân    '''
17 S=Bai64(S)
18 print(S)
19

```

temp

Bai07\_TR27

temp (1)

Bai64\_Tr14

Python 3.9.5 (tags/v3.9.5:0a7dcdbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]

In[3]:

01 S = {str}'

Special V

Buoi06 &gt; Bai64\_Tr14.py

Project temp.py Bai64\_Tr14.py Bai6\_Tr26.py Bai07\_TR27.py

1: Project  
▼ Buoi06 D:\TTTH\_KHTN\Python270  
    Bai6\_Tr26.py  
    Bai07\_TR27.py  
    Bai64\_Tr14.py  
    temp.py  
▶ External Libraries  
Scratches and Consoles

```
14  quê hương là đường đi học .  
15  Con về rợp bướm vàng bay.  
16  Đố Trung Quân  
17  S=Bai64(S)  
18  print(S)  
19
```

temp Bai07\_TR27 temp (1) Bai64\_Tr14

```
sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi06', 'D:/TTTH_KHTN/Python270/Buoi06'])
```

```
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]
```

```
Quê hương
```

```
Quê hương là chùm khế ngọt.
```

```
Cho con trèo hái mỗi ngày.
```

```
Quê hương là đường đi học.
```

```
Con về rợp bướm vàng bay.
```

```
Đố Trung Quân
```

```
In[3]:
```

F2							9
	A	B	C	D	E	F	G
1		0	1	2	3	4	
2		7	4	5	3	9	
3							

```
def TongKC(lst):  
    Tong = 0  
    for i in range(0, len(lst)-2):  
        for j in range(i+1, len(lst)-1):  
            Tong = Tong + abs(lst[i]-lst[j])  
    return Tong
```

#-----

28/. Viết chương trình nhập vào năm. In ra tên của năm âm lịch tương ứng.

Ví dụ: nhập năm 2019 in ra *Kỷ Hợi*.

Biết rằng:

<b>CAN</b>	<i>Giáp</i>	<i>At</i>	<i>Bính</i>	<i>Đinh</i>	<i>Mậu</i>	<i>Kỷ</i>	<i>Canh</i>	<i>Tân</i>	<i>Nhâm</i>	<i>Quý</i>		
<b>CHI</b>	<i>Tý</i>	<i>Sửu</i>	<i>Dần</i>	<i>Mão</i>	<i>Thìn</i>	<i>Tỵ</i>	<i>Ngọ</i>	<i>Mùi</i>	<i>Thân</i>	<i>Dậu</i>	<i>Tuất</i>	<i>Hợi</i>

project temp.py Bai28\_Tr31.py Bai22\_Tr30.py Bai64\_Tr14.py Bai10\_Tr27.py Bai6\_Tr26.py Bai07\_TR27.py

Buoi06 D:\TTTH\_KHTN\Python270

- Bai6\_Tr26.py
- Bai07\_TR27.py
- Bai10\_Tr27.py
- Bai22\_Tr30.py
- Bai28\_Tr31.py
- Bai64\_Tr14.py
- temp.py

External Libraries

Scratches and Consoles

```
1 Can=["Canh","Tân","Nhâm","Quý","Giáp","Ất","Bính","Đinh","Mậu","Kỷ"]
2 Chi=["Thân","Dậu","Tuất","Hợi","Tý","Sửu","Dần","Mão","Thìn","Tỵ","Ngọ","Mùi"]
3
4 nam=int(input("Nhập năm dương lịch: "))
5 print("Năm %d là năm %s %s" %(nam, Can[nam%10], Chi[nam%12]))
```

*thuật toán này:*

- > đẩy Canh về vị trí index 1 (vì có 10 số tương ứng với Can mà đầu tiên là Canh)
- > đẩy Thân về vị trí index 1 (vì có 12 năm tương ứng với Chi mà đầu tiên là Thân)

mp Bai07\_TR27 temp (1) Bai64\_Tr14 Bai28\_Tr31

```
"C:\Program Files\Python39\python.exe" "C:\Program Files\JetBrains\PyCharm Community I"
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi06', 'D:/TTTH_KHTN/Python270/Buoi06'])
```

Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]

Nhập năm dương lịch:

>?

Special Variables

```
project ▾ | temp.py | Bai28A.py | Bai28_Tr31.py | Bai22_Tr30.py | Bai64_Tr14.py | Bai10_Tr27.py | Bai6_Tr26.py | Bai07_TR27.py
Bui06 D:\TTTH_KHTN\Python270
Bai6_Tr26.py
Bai07_TR27.py
Bai10_Tr27.py
Bai22_Tr30.py
Bai28_Tr31.py
Bai28A.py
Bai64_Tr14.py
temp.py
External Libraries
Scratches and Consoles

1 # nhập số. In ra chữ số tương ứng 9452, 21, 789654312
2 def InChuSo(n):
3     ChuSo=['không', 'một', 'hai', 'ba', 'bốn', 'năm', 'sáu', 'bảy', 'tám', 'chín']
4     S=int(str(n)[::-1])
5     print(S)
6
7 InChuSo(123)    thuật toán đảo chiều string

InChuSo()
```

```
mp | Bai07_TR27 | temp (1) | Bai64_Tr14 | Bai28_Tr31 | Bai28A | Special Variables
"C:\Program Files\Python39\python.exe" "C:\Program Files\JetBrains\PyCharm Community
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Bui06', 'D:/TTTH_KHTN/Python270/Bui06'])
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]
321
In[3]:
```



Buoi06 &gt; Bai28A.py

```
Project
1: Project
  Buoi06
    Bai6_Tr26.py
    Bai07_TR27.py
    Bai10_Tr27.py
    Bai22_Tr30.py
    Bai28_Tr31.py
    Bai28A.py
    Bai64_Tr14.py
    temp.py
  External Libraries
  Scratches and Consoles

temp.py x Bai28A.py x Bai28_Tr31.py x Bai22_Tr30.py x Bai64_Tr14.py x Bai10_Tr27.py x Bai6_Tr26.py x Bai07_
1 # nhập số. In ra chữ số tương ứng 9452, 21, 789654312
2 def InChuSo(n):
3     ChuSo=['không','một','hai','ba','bốn','năm','sáu','bảy','tám','chín']
4     if n<0:
5         print('am ', end=' ')
6         n=-1*n
7         so=int(str(n)[::-1])
8     while so>0:
9         kq=so%10
10        print('%s' % ChuSo[kq], end=' ')
11        so=so//10
12
13 InChuSo(-123456789)

InChuSo()
```

```
temp x Bai07_TR27 x temp (1) x Bai64_Tr14 x Bai28_Tr31 x Bai28A x
sys.path.extend(['D:\\TTTH_KHTN\\Python270\\Buoi06', 'D:/TTTH_KHTN/Python270/Buoi06'])
```

```
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)]In[2]: runfile('
am một hai ba bốn năm sáu bảy tám chín
In[3]:
```





1: Project

Buoi06 D:\TTTH\_KHTN\Python270

Bai6\_Tr26.py

Bai07\_TR27.py

Bai10\_Tr27.py

Bai22\_Tr30.py

Bai28\_Tr31.py

**Bai28A.py**

Bai64\_Tr14.py

temp.py

External Libraries

Scratches and Consoles

Project

temp.py

**Bai28A.py**

Bai28\_Tr31.py

Bai22\_Tr30.py

Bai64\_Tr14.py

Bai10\_T

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

```

if n<0:
    print('am ', end=' ')
    n=-1*n
    #Cách 1: đảo chuỗi
    '''so=int(str(n)[::-1])
    while so>0:
        kq=so%10
        print('%s' % ChuSo[kq], end=' ')
        so=so//10
    ...
    # Cách 2: không đảo
    k=1
    while k*10<n:
        k=k*10
    while k>0:
        kq=n//k
        print('%s' % ChuSo[kq], end=' ')
        n=n%k
        k=k//10
    #-

```